

↓ step 1 Chapter 5.11

a) Given data  
virtual address = 32 bits  
PTE size = 4 bytes

$$\begin{aligned}\text{Page table entries (PTE)} &= \text{Virtual address} - \text{page size} \\ &= 32 - 13 \text{ (8 KB)} \\ &= 19 \text{ bits or 512 entries}\end{aligned}$$

step 2

$$\begin{aligned}\text{Physical memory} &= \text{PTE} \times \text{PTE size} \\ &= 512 \times 4 \\ &= 2048 \text{ or } 2 \text{ MB}\end{aligned}$$

step 3

ⓑ Given data:  
virtual address = 64 bits

Page size = 4 KB

PTE size = 8 bytes

$$\begin{aligned}\text{Page table entries (PTE)} &= \text{Virtual address} - \text{page size} \\ &= 64 - 12 \text{ (4 KB)} \\ &= 52 \text{ bits or } 2^{52} \text{ entries}\end{aligned}$$

step 4

$$\begin{aligned}\text{Physical memory} &= \text{PTE} \times \text{PTE size} \\ &= 2^{52} \times 2^3 \\ &= 2^{55} \text{ bytes}\end{aligned}$$

2 from S. 11

a) step 1  
Given data

Virtual address = 32 bits

Page size = 8 KB

PTE size = 4 bytes

Page table entries (PTE) = Virtual address - page size

$$= 32 - 13 (8 \text{ KB})$$

$$= 19 \text{ bits or } 512 \text{ entries}$$

step 2

8 KB page / 4 byte PTE =  $2^9$  pages indexed per page

Hence with  $2^{19}$  PTEs will need 2-level page setup

Each address translation will require at least

2 physical memory access.

step 3

b) Virtual address = 64 bits

page size = 4 KB

PTE size = 8 bytes

Page table entries (PTE) = Virtual address - page size

$$= 64 - 12 (4 \text{ KB})$$

$$= 52 \text{ bits or } 2^{53} \text{ entries}$$

step 4

4 KB page / 8 byte PTE =  $2^9$  pages indexed per page

Hence with  $2^{52}$  PTEs will need 6-level page table setup

Hence address translation will require at least 6 physical

memory access

3 from  
a) steps  
Given data

5.11

virtual address = 32 bits

Physical DRAM = 4GB

Page size = 8KB and

PTE size = 4 byte

Number of PTE =  $32 - 24$   
= 8 bits

step 2

Therefore 512K PTE's are needed to store the page table

In common case: no hash conflict, so one memory reference per address translation.

In worst case: 512 memory references are needed if hash table degrades into a link list.

step 3

(b) Given data

virtual address = 64 bits

Physical DRAM = 16GB

Page size = 4KB

PTE " = 8 byte

Number of PTE =  $64 - 12$   
= 52 bits

Therefore  $2^{(34-12)}$  PTE's are needed to store the page table

In common case: no hash conflict. So one memory reference per address information

In worst case:  $2^{(34-12)}$  memory references are needed if hash table degrades into link list